

## **REMARKS**

**[0003]** Applicant respectfully requests reconsideration and allowance of all of the claims of the application. Claims 1, 2, 4-26 are presently pending. Claim 3 is cancelled without prejudice or disclaimer. Claims amended herein are 1, 2, 4-11, and 16-22.

### **Statement of Substance of Interview**

**[0004]** The Examiner graciously talked with me—the undersigned representative for the Applicant—on August 28, 2007. Applicant greatly appreciates the Examiner's willingness to talk. Such willingness is invaluable to both of us in our common goal of an expedited prosecution of this patent application.

**[0005]** During the interview, I discussed the features in the proposed amendment and how the proposed claims differed from the cited art. Without conceding the propriety of the rejections and in the interest of expediting prosecution, I also proposed several possible clarifying amendments.

**[0006]** Applicant herein amends the claims in the manner discussed during the interview. Accordingly, Applicant submits that the pending claims are allowable over the cited art of record for at least the reasons discussed during the interview.

### **Formal Request for an Interview**

**[0007]** If the Examiner's reply to this communication is anything other than allowance of all pending claims, then I formally request an interview with the Examiner. I encourage the Examiner to call me—the undersigned representative for the Applicant—

so that we can talk about this matter so as to resolve any outstanding issues quickly and efficiently over the phone.

**[0008]** Please contact me or my assistant to schedule a date and time for a telephone interview that is most convenient for both of us. While email works great for us, I welcome your call to either of us as well. Our contact information may be found on the last page of this response.

### **Claim Amendments**

**[0009]** Without conceding the propriety of the rejections herein and in the interest of expediting prosecution, Applicant amends claims 1, 2, 4-11, and 16-22 herein.

**[0010]** Independent claims 1, 16, and 19 are amended to recite, *inter alia*, “a web software application that contains one or more resources...” Support for the amendment can be found throughout the Application including, for example, Fig. 2 with the associated text (“The App 201... includes a group of resources 211 that are linked together such that a user may navigate from one resource to another. The resources 211 are typically Web pages..., but may also include other resources, such as code, images, or the like.” Specification at p.7 lines 19-23).

**[0011]** Claims 1, 16, and 19 are further amended to recite, *inter alia*, “wherein states of the web software application are persisted in an execution environment during the execution and made accessible via run-time objects to the resources of the web software application...” Support for the amendment can be found throughout the Application including, for example, Fig. 2 with the associated text. In particular, “[a]ny resource within the application may preserve state in memory through the use of a

NavigationApplication object 205.” (Specification at p.8, lines 203). “[T]he App 201 is hosted in some sort of execution environment in which the resources 211 are interpreted or executed. The execution environment may be Web browsing software or other execution environments, .. The NavigationApplication object 205 serves as the entry point of the App 201 and is used to define characteristics and parameters of the App 201...” *Id.* at p.8, lines 5-10. “Any of the resources 211 that need or desire state information about the application include instructions that access the NavigationApplication object 205 for that state information.” *Id.* p.9, lines 2-4.

**[0012]** Furthermore, claim 16 is amended to recite, *inter alia*, “a StartingUp method including executable instructions to be executed to load state of the web software application when it is being launched...” Support for the amendment can be found throughout the Application including, for example, Fig. 5 with the associated text. In particular, “[t]he several events 307 of the application class 301 include a StartingUp event 318 and a ShutDown event 319. The application developer can attach a handler to the StartingUp event 318 to initialize the application or open a new window and present the user with some user interface...” (Specification at p.12, lines 7-10). “[T]hose two events could be used in conjunction to persist state on shutdown and to restore it on startup. This technique may be used to create the impression that the application’s state remains just as it was when the user left it” *Id.* at p.12, lines 13-16.

**[0013]** Accordingly, no new matter will be introduced by the amendment. Entry to the file is respectfully requested.

## **Substantive Matters**

### **Claim Rejections under § 101**

**[0014]** Claims 1-26 are rejected under 35 U.S.C. § 101. In light of the amendments presented herein, Applicant respectfully submits that these claims comply with the patentability requirements of § 101 and that the § 101 rejections should be withdrawn. The Applicant further asserts that these claims are allowable. Accordingly, Applicant asks the Examiner to withdraw these rejections.

**[0015]** If the Examiner maintains the rejection of these claims, then the Applicant requests additional guidance as to what is necessary to overcome the rejection.

### **Claim Rejections under §§ 102 and/or 103**

**[0016]** Claims 1-26 are rejected under 35 U.S.C. § 102 and/or § 103 for being unpatentable over U.S. Patent No. 6,810,395 to Bharat ("Bharat") and U.S. Patent No. 6,684,383 to Natori et al. ("Natori"). In light of the amendments presented herein and the decisions/agreements reached during the above-discussed Examiner interview, Applicant submits that these rejections are moot. Accordingly, Applicant asks the Examiner to withdraw these rejections.

**[0017]** Independent claim 1, as amended, recites (Emphasis added):

1. A software architecture implemented at least in part by a computing device for executing a web software application that contains one or more resources, comprising:

*a first set of application programming interfaces*, when implemented and executed by the computing device, configured to support

the execution of the web software application within the software architecture; and

*a second set of application programming interfaces, when implemented and executed by the computing device, configured to support navigation-related activities of the web software application,*

*wherein states of the web software application are persisted in an execution environment during execution and made accessible via run-time objects to the resources of the web software application by the first and second sets of application programming interfaces.*

**[0018]** Claim 1 stands rejected under 35 U.S.C. §102 for being anticipated by Bharat. Applicant herein respectfully submits that not all features in the amended claim 1 are taught or disclosed in Bharat. In particular, neither “application programming interface” (“API”) nor the feature that “states of the web software application are persisted in an execution environment... and made accessible via run-time objects to the resources of the web software application...” are present in Bharat.

**[0019]** Bharat is directed to a method and system providing a query-specific bookmarking and query-specific data collection to maintain the search context when a user browses the Internet. As shown in Fig. 6 of Bharat, a query is passed to a search engine, which then returns the query result page to the browser/user. When the user selects and marks the leads he wants to save, such leads are then stored and/or sent to SearchPad (a Leads marking software). (See also Fig. 8(b)).

**[0020]** In rejecting claim 1, the Office Action takes the position that Bharat teaches a first set of APIs that support the execution of the application within the software architecture (web application, abstract and col. 1-2), and a second set of APIs that support navigation-

related activities of the software application (Brower, figures, 6, 7a&b, 8a&b, and 10).

Applicant respectfully traverses the rejection.

**[0021]** First, it's respectfully asserted that neither the query-specific book-marking software nor the browser software is the Application Programming Interface referred to in amended claim 1 (Emphasis added). An API "refers to a source code interface that a computer application, operating system or library provides to support requests for services to be made of it by a computer program." (See [www.wikipedia.org](http://www.wikipedia.org)). "An API is typically defined at a higher level (i.e., in terms of a programming language that can be compiled when an application is built." *Id.* "The API itself is abstract, in that it specifies an interface and does not get involved with implementation details." *Id.*

**[0022]** The API recited in claim 1 is well supported in the Application including, for example, Figs. 3 and 4. Bharat, on the other hand, is completely silent in teaching the APIs with respect to either the browser or the query bookmarking software. At best, Bharat discloses a query bookmarking software, either independent of web browser or incorporated with web browser (Bharat, Figs. 7a&b), that facilitates the recording of specific web pages with a provided query. Bharat, however, does not disclose a software APIs in code or abstract level.

**[0023]** Secondly, the query bookmarking software in Bharat, as referred to in the Office Action as "a first set of API ", does not "support the execution of the web software application within the software architecture." In fact, as disclosed in Bharat, the query bookmarking software is designed to "save/send query-specific bookmarks (and any other query-specific information) as leads in association with the query." (See Bharat, Fig. 1). Bharat further mentions that the browser and query bookmarking software are separate (See

Bharat, Fig. 7a). There is nowhere in Bharat that mentions the query bookmarking software actually “supports the execution of the web software application...” Applicant respectfully submits that this feature in claim 1 is totally missing in Bharat.

**[0024]** Furthermore, the feature “states of the web software application are persisted in an execution environment during the execution and made accessible via run-time objects to the resources...” is missing in Bharat too. According to Bharat, after the query result is received, the query bookmarking software “[saves/sends] query-specific bookmarking (and any other query-specific information) as leads” as, for example, a cookie by a user. Accordingly, the leads in Bharat are actually not “persisted in an execution environment,” nor are they “made accessible via run-time objects to the resources...” as recited in amended claim 1. Therefore, Bharat does not teach or disclose the amended feature at all.

**[0025]** It’s asserted that amended claim 1 is patentably distinct from Bharat. Similarly, the feature “Properties collection” and “StartUpURI property” being “persisted in an execution environment and made accessible via run-time objects to the resources...”, as recited in claim 19, is missing in Bharat too. Thus, claim 19 is also asserted patently distinct from Bharat.

**[0026]** Claim 16 stands rejected under 35 U.S.C. §102 for being anticipated by Natori. Applicant amends claim 16 herein, rendering the rejection moot.

**[0027]** Claim 16, as amended, recites:

16. A computer-readable medium having computer-executable components for supporting the execution of a web software application that contains one or more resources, the components comprising:

an application programming interface exposed by the software application, the application programming interface including:

a StartingUp method including executable instructions to be executed to load states of the web software application when it is being launched; and

a ShutDown method that, when called, is operative to cause the states of the web software application to be saved before it is shut down,

wherein the states of the web software application are *persisted in an execution environment during execution of the web software application and made accessible via run-time objects to the resources of the web software application by the application programming interface.*

**[0028]** Applicant herein submits that Natori does not teach or disclose at least “states of the web software application are persisted in an execution environment ...and made accessible via run-time objects to the resources of the web software application...”

**[0029]** Natori is directed to an enterprise framework comprising web application system framework. (See Natori, Fig. 1). The web application system framework, as defined in Natori, “[i]nherits the enterprise system basic framework 11. The web application system framework 13 is a software skeleton which abstractly defines basic attributes and behaviors of a network-oriented client/server application system, and is expressed as an aggregate of abstract classes and concrete classes.” (Natori, col. 8, lines 3-9). Furthermore, “the web application system framework 13 provides basic functions which relate to the start and end of systems, the delivery of data between systems, the transmission and acquisition of requests, the input/output of data to systems, the transition between systems, system



control, and connection interfaces to common components, in a system which is realized on many and unspecified clients 31...” (Natori, col. 8, lines 12-22).

**[0030]** However, the feature recited in amended claim 16 is absent in Natori. Natori does disclose to compile / link source code for constructing an enterprise system. As shown in Fig. 20, sub-frameworks (the client/server application system framework 12, the Web application system framework 13, the server application system framework 14 and the framework for integrating system 15) inherit the enterprise system basic framework, and the customized implementations of the relevant methods for each of the sub-frameworks override the basic methods such that one or more desired functions are thus available. “Finally, the source code thus extended is compiled and linked to generate an executable module which is executable on a computer system” (Natori, col. 19, lines 1-8).

**[0031]** It is asserted that construction of the enterprise system in Natori does not teach all the features recited in claim 16. In particular, Natori is completely silent in persisting states of a web software application in an execution environment (Natori only teaches to compile source code to executable module) and silent in making the states of application accessible to the resources of a web software application via run-time objects.

**[0032]** Therefore, Applicant submits that amended claim 16 is patentably distinct from Natori.

### **Dependent Claims**

**[0033]** In addition to its own merits, each dependent claim is allowable for the same reasons that its base claim is allowable. Applicant requests that the Examiner withdraw the rejection of each dependent claim where its base claim is allowable.

## CONCLUSION

**[0034]** All pending claims are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the application. If any issues remain that prevent issuance of this application, the Examiner is urged to contact me before issuing a subsequent Action. Please call/email me or my assistant at your convenience.

Respectfully Submitted,

Dated: 9/10/07

By: Kayla Brant

Kayla Brant  
Reg. No. 46576  
(509) 324-9256 x242  
kayla@leehayes.com  
[www.leehayes.com](http://www.leehayes.com)

My Assistant: Carly Bokarica  
(509) 324-9256 x264  
[carly@leehayes.com](mailto:carly@leehayes.com)